

IN THE UNITED STATES DISTRICT COURT
FOR THE SOUTHERN DISTRICT OF TEXAS
HOUSTON DIVISION

United States Courts
Southern District of Texas
ENTERED

SEP 02 2003

AMERICAN IMAGING
SERVICES, INC.,

Plaintiff,

VS.

INTERGRAPH CORP.,

Defendant.

§
§
§
§
§
§
§
§
§

CIVIL ACTION NO. H-97-1394

Michael H. Milby, Clerk of Court

MEMORANDUM AND OPINION

American Imaging Services, Inc. alleges that Intergraph Corp.'s products infringed U.S. Patent No. 5,353,393 to Bennett *et al.* (the "'393 patent"). This court previously granted Intergraph's motion for summary judgment on the basis that the claims in the '393 patent were anticipated and obvious, making them invalid under 35 U.S.C. §§ 102, 103. American Imaging appealed this court's decision to the Federal Circuit, which affirmed in part, reversed and vacated as to four of the claims in the patent, and remanded to this court. Intergraph then moved for summary judgment as to the four remaining claims, asserting that they were obvious and therefore invalid under 35 U.S.C. § 103. (Docket Entry No. 69). This court denied Intergraph's motion. (Docket Entry No. 84). Intergraph now moves for summary

judgment on a different ground, asserting that its software does not infringe the remaining claims of the '393 Patent and that those claims are invalid under 35 U.S.C. §§ 102(b) and (g). Intergraph also seeks to strike the supplemental expert report of Gordon Peterson that American Imaging included in its response to Intergraph's motion for summary judgment. (Docket Entry No. 131; Docket Entry No. 128 Ex. B, Declaration of Gordon Peterson ("Peterson Decl.")).

Based on a careful review of the motion and response, the parties' submissions, the Federal Circuit's decision, and the applicable law, this court grants Intergraph's motion for summary judgment based on noninfringement. This court grants Intergraph's motion to strike Peterson's declaration, but alternatively holds that even if the declaration is admitted and considered as summary judgment evidence, summary judgment is proper. The reasons for these rulings are set out below.

I. Background

A. Procedural History

The '393 patent relates to an apparatus and method for manipulating scanned documents using computer-aided design ("CAD") commands. The technology described in the '393 patent allows conversion of a paper document, such as a hand-drawn engineering design, into computer-readable electronic form; permits changes to the document using computer tools; and makes the edited version

available in both hard copy and electronic forms. Intergraph manufactures a family of software called “I/RAS,” which it developed in coordination with Bentley Systems, Inc. This software allows raster graphics to be manipulated in “MicroStation,” a CAD program that Bentley manufactures. (Docket Entry No. 114, Declaration of Curtis A. Lipkie (“Lipkie Decl.”), ¶ 9).

American Imaging, the holder of the ‘393 patent, sued Intergraph for infringement. This court granted Intergraph’s motion for summary judgment on the grounds that the ‘393 patent claims were anticipated and obvious in light of the “SuperPaint” software, a general graphics paint program that had entered the market two years before the applicants filed for the ‘393 patent. American Imaging appealed that decision to the Federal Circuit, which affirmed in part, reversed in part, and remanded for further proceedings. The Federal Circuit affirmed this court’s finding of anticipation as to claims 1 through 7 and claims 14 through 29. Although the Federal Circuit disagreed with this court’s findings of anticipation as to claims 8 through 11, 30 through 34, and 37 through 41, because those claims limited the electronic display means to a CAD system, which was not present in the SuperPaint software, the Federal Circuit found that those claims were obvious in light of

SuperPaint and affirmed this court's finding of invalidity on that basis.¹ The Federal Circuit explained:

The '393 patent sought to solve the problem of manipulating scanned documents using CAD commands and facilitating the editing of a document and its reproduction in hard copy form. *See* '393 Patent, Col. 2, ll. 1-17. Although SuperPaint does not operate to produce engineering drawings, it teaches the art of scanning an image to create a raster file, modifying the raster image using vector commands and merging and editing the raster image and the vector changes in a paint program that can be reproduced in hard copy form. More pointedly, SuperPaint teaches, in a general graphics program, to one of ordinary skill in the art of computer programming, the solution sought by the '393 patent. It would have been well within the knowledge of one of ordinary skill in the art to apply these teachings within a CAD system, which itself is a species of graphics programs. Therefore, we conclude that claims 8 through 11, 13, 30 through 34, and 37 through 41 would have been obvious as a matter of law in light of SuperPaint and the knowledge generally available to one of ordinary skill in the art attempting to solve the problem these claims sought to address.

(Docket Entry No. 58, pp. 12-13).

The Federal Circuit reversed this court's finding of obviousness as to claims 12, 35, 36, and 42, stating as follows:

[A]bsent a teaching, suggestion or motivation that one of

¹ The Federal Circuit defined a "CAD system" as "a species of graphics programs particularly well-suited for producing engineering drawings." (Docket Entry No. 58, p. 7). The court found that SuperPaint did not contain this limitation. (*Id.*).

ordinary skill in the art would reconfigure SuperPaint for use in an operating system having an interrupt vector table, [the] conclusion of obviousness is erroneous.

Furthermore, the record lacks any evidence regarding whether SuperPaint, which admittedly does not include the ability to operate with an interrupt vector table, suggests to one of ordinary skill in the art to reconfigure SuperPaint to enable it to work with an operating system having an interrupt vector table. Although an expressed suggestion to configure SuperPaint to work with such an operating system is not required, there must be some suggestion found in the field of knowledge generally available to one of ordinary skill in the art or motivation to combine from the problem itself. [citations omitted]. Mr. Snider's declaration, quoted by the district court in support of its obviousness determination is merely a conclusory statement that one could write a program to address these matters. His declaration, especially because he was not versed in DOS operating systems, is insufficient to support a conclusion of obviousness.

Moreover, SuperPaint only teaches the art of converting a document into computer-readable form wherein the user can modify the document and reproduce it in hard-copy form. SuperPaint does not teach one of ordinary skill in the art the ability to reconfigure the SuperPaint program to operate within a conventional DOS operating system and the problem these claims sought to resolve would not motivate an ordinarily skilled artisan to reconfigure a Macintosh-compatible paint program to manipulate an interrupt vector table. Because there are genuine issues of material fact that must be resolved in order to determine whether claims 12, 35, 36 and 42 would have been obvious, summary judgment on these claims was incorrect.

(Docket Entry No. 58, pp. 13-14).

On remand, Intergraph has again moved for summary judgment and has provided additional evidence. Intergraph now asserts that its I/RAS software does not infringe the '393 Patent and that claims 12, 35, 36, and 42 are invalid under sections 102(b) and (g) based on the development, sale, and use of a program called CAD Overlay, which allowed scanned documents to be used with a CAD software package known as AutoCAD.

Intergraph submits the declarations of Steven Entine, Chief Technology Officer of Wingra Technologies, LLC; Curtis A. Lipkie, the principal designer of I/RAS; and David Chassin, one of the developers of CAD Overlay; excerpts from MicroStation manuals; and copies of I/RAS source code. American Imaging has responded and submitted additional evidence of its own, including the declaration of Gordon Peterson, a computer software expert retained by American Imaging; excerpts from MicroStation manuals; and copies of I/RAS source code.

B. The '393 Patent Claims

The four claims before this court describe a system that allows a CAD system, which typically displays only vector images, to interact with an auxiliary program that permits the user to view the CAD system's vector image while simultaneously viewing and editing a raster image.

Claim 12 claims:

Apparatus according to claim 8,² wherein said electronic data processing system includes an operating system having an interrupt vector table, further comprising:

driver identification means for identification of existence of a display driver interrupt address in the vector table,

relocation means for relocating a CAD generated display driver interrupt from said display interrupt address to a clear location, and

remapping means for remapping a link program means to said display driver interrupt address.

(Docket Entry No. 110, Ex. E ('393 Patent), col. 11, 1.10-22).

Claim 35 claims:

Apparatus according to claim 31,³ wherein said CAD system includes an operating system having an interrupt vector table, further comprising:

driver identification means for identification of existence of a display driver interrupt address in the vector table,

relocation means for relocating a CAD generated display driver interrupt from said display driver interrupt address to a clear location, and

² The contents of claim 8 and claim 1, on which claim 8 depends, are not critical to the infringement analysis.

³ The contents of claim 31 and claim 30, on which claim 31 depends, are not critical to the infringement analysis.

remapping means for remapping a link program means to said display driver interrupt address.

(*Id.* at col. 15, l.27-38).

Claim 36 claims:

Apparatus according to claim 35, further comprising intercepting means for intercepting a display driver interrupt generated by said CAD system in response to a user input command, and

service means for servicing said interrupt by accessing and editing raster data in a raster data base and by driving the computer screen display in accordance with the edited data from the raster data base.

(*Id.* at col. 15, l.39-col. 16, l.2).

Claim 42 claims:

Apparatus according to claim 37 wherein said edit program means comprises:

driver identification means for identification of existence of a display driver driven by the CAD system,

relocation means for relocating a display driver identified by said driver identification means from a driver address to a clear location therefor, and

remapping means for remapping the edit program means to said driver address.

(*Id.* at col. 16, l.34-l.43).

The claimed system interacts with the auxiliary raster editing program by redirecting program flow through use of an “interrupt vector table.” An “interruption” means that “one program is stopped, another executed, and the former resumed, without any effect except for a time delay, upon the executing program.” (Docket Entry No. 72, Ex. B, Walter Rudd, ASSEMBLY LANGUAGE PROGRAMMING AND THE IBM 360 AND 370 COMPUTERS 234 (Prentice Hall 1976); Docket Entry No. 102, ¶ 1). A number of operating systems manage interrupts by using an “interrupt vector table,” a register that contains the addresses of numerous interrupt service routines. (Docket Entry No. 72, Ex. C, Andrew S. Tanenbaum, STRICTURED COMPUTER ORGANIZATION, 3d Ed. 124 (Prentice Hall 1990); Ex. F, George Gorsline, MODERN MICROCOMPUTERS: THE INTEL I8086 FAMILY 159-60 (Prentice Hall 1985); Docket Entry No. 102, ¶ 1). An I/O controller chip of the computer determines which device caused the interrupt. (Docket Entry No. 72, Ex. C, Andrew S. Tanenbaum, STRICTURED COMPUTER ORGANIZATION, 3d Ed. 124 (Prentice Hall 1990)). The computer then searches the interrupt vector table for the address of the interrupt service routine to run to service the interrupt. (*Id.*).

Claims 12, 35, and 36 describe the redirection or “remapping” of the CAD system’s display driver interrupt to a “link” program, which displays a raster image and allows the user to overlay the CAD system’s vector image on the raster

image. Claim 42 describes the remapping of the CAD display driver interrupt to an “edit” program, which allows the user to modify the raster image. Together, the “link” and “edit” programs permit the user simultaneously to view a vector image and to view and edit a raster image. The “link” and “edit” programs are accessory programs that can operate with an unmodified version of the CAD system to achieve the added benefit of simultaneously viewing and editing raster images. The “link” and “edit” software allows the user to merge the vector-created modifications into the raster image.

In a typical CAD system, the user inputs commands through a device connected to the CAD system. The CAD system generates commands to a display driver, which processes the commands and sends the output to a hardware display device. The display device, such as a computer screen, displays the CAD drawing. (‘393 Patent, col 4, 1.9-1.17). The typical CAD system accomplishes this by generating a display driver interrupt that calls the interrupt vector table address of the display driver interrupt routine. The interrupt service routine for the display driver then runs, and the computer screen (or other output device) displays, the user’s input.

The claimed system incorporates the link or edit program of claims 12, 35, 36, and 42 with the CAD system and the display driver. The link program determines whether there is a valid display driver address in the interrupt vector table.

The link program moves the interrupt service routine commanding the display driver to a new address in the interrupt vector table. The link program is then mapped to the display driver's original interrupt address. ('393 Patent, col. 5, 1.61-col. 6, 1.9). When the CAD system generates a display driver interrupt that calls the display driver's original interrupt address, the link or edit program is the interrupt service routine. The link or edit program services the display driver interrupts generated by the CAD system. If the user's input is a change to the raster image stored in the link or edit program, that program will make the change and send the output to the display driver. The driver display interrupt service routine then operates on the output of the link or edit programs and the results appear on the computer screen. (*Id.* at Fig. 2A). If the user's input does not involve a change to the raster image, the input bypasses the link or edit program and is sent directly to the display driver. (*Id.* at col. 6, 1.18-1.27).

The written description of the '393 Patent contains an alternative embodiment of the invention in which the link or edit programs are not interposed between the CAD system and the display driver. Instead, the CAD system is directly connected to the link or edit programs through an "interprocess communication path (e.g., a pipe)" and issues commands directly to the raster editor. (*Id.* at col. 9, 1.8-1.23).

C. The Accused Device

The I/RAS software family was designed to add raster editing functionality to MicroStation, a CAD software package written by Bentley Systems, Inc. I/RAS communicates with MicroStation through a process called “dynamic linking.” (Entine Decl., ¶¶ 14-16). Dynamic linking allows one computer program to make calls to functions in a different computer program. (*Id.* at ¶ 16). I/RAS and MicroStation are separate programs written in a C-based computer language called MicroStation Development Language (“MDL”). (Lipkie Decl., ¶ 18). I/RAS consists of a main routine—known as “main”—and several computer subroutines that call, and can be called by, main. Some of I/RAS’s subroutines call subroutines in the MicroStation CAD system through dynamic linking. (Lipkie Decl., ¶¶ 23-25, 28). I/RAS’s main routine and subroutines can call several subroutines within MicroStation, including “mdlView_setFunction,” which sets a function to handle particular events, such as a window update, and “mdlWindow_lineDraw,” which draws lines on the screen. (Entine Decl., Ex. 2, pp. 1-2; Ex. 3 at B06315, B06696; ING-025388.025, ING-025388.103). I/RAS requests a call back from MicroStation whenever MicroStation is about to make a screen update, so that I/RAS can send to MicroStation the raster image to be displayed on the computer screen. (*Id.* at ¶ 24).

For example, I/RAS's output subroutines could call "mdlWindow_lineDraw," a function within MicroStation that plots lines on the computer screen.⁴

The raster image to be manipulated is stored in I/RAS. A user can edit the raster image with vector drawings in MicroStation. When a user makes vector edits to a raster drawing and the screen display is to be updated, I/RAS subroutines determine the correct area of a raster file to send to MicroStation. (Lipkie Decl. at ¶¶ 27). Some of these I/RAS subroutines call the MicroStation function "mdlWindow_lineDraw," which draws the selected portion of the raster file on the computer screen. (*Id.*). Because I/RAS and MicroStation are coordinated, the raster image from I/RAS and the vector image from MicroStation are positioned in concert on the computer screen, creating a hybrid raster/vector image. (*Id.* at ¶¶ 26, 28).

The issue raised by this summary judgment motion is infringement, not validity. Intergraph asserts that the undisputed facts in the summary judgment record

⁴ The detailed operation of I/RAS is as follows: I/RAS calls MicroStation subroutine mdlView_setFunction(UPDATE_PRE, irasb_hook_update). This makes MicroStation call I/RAS function "irasb_hook_update" whenever MicroStation is about to update the image on the computer screen by pointing the pointer associated with the variable UPDATE_PRE to the memory address of the function irasb_hook_update. (Entine Decl., ¶ 26, Ex. 3, ING-025388.023 at 1.167; Ex. D at B06696). The function irasb_hook_update calls the I/RAS function "red_view_driver," where "red" stands for "raster editor." (*Id.* at ING-025388.035, 1.214). The function red_view_driver calls the I/RAS function "red_view_driver_2." (*Id.* at ING-025388.055, 1.746, ING-025388.057, 1.820). The function red_view_driver_2 calls the I/RAS function "red_plot_line" in several places. (*Id.* at, for example, ING-025388.073, 1.1545; ING-025388.075, 1.1599, 1.1627). The function red_plot_line calls the MicroStation function "mdlWindow_lineDraw," which draws the line on the computer screen. (*Id.* at ING-025388.103, 1.2814). (Docket Entry No. 110, Ex. D).

establish that as a matter of law, its I/RAS software neither literally infringes the ‘393 Patent nor infringes under the doctrine of equivalents.

II. The Summary Judgment Standard

Summary judgment is appropriate if no genuine issue of material fact exists and the moving party is entitled to judgment as a matter of law. *See* FED. R. CIV. P. 56. Under FED. R. CIV. P. 56(c), the moving party bears the initial burden of “informing the district court of the basis for its motion, and identifying those portions of [the record] which it believes demonstrate the absence of a genuine issue of material fact.” *Celotex Corp. v. Catrett*, 477 U.S. 317, 323 (1986); *Norman v. Apache Corp.*, 19 F.3d 1017, 1023 (5th Cir. 1994). The party moving for summary judgment, whether a plaintiff or defendant, must demonstrate the absence of a genuine issue of material fact, but need not negate the elements of the nonmovant’s case. *See Little v. Liquid Air Corp.*, 37 F.3d 1069, 1075 (5th Cir. 1994) (en banc). If the moving party fails to meet its initial burden, the motion for summary judgment must be denied, regardless of the nonmovant’s response. *See id.*

When the moving party has met its Rule 56(c) burden, the nonmovant cannot survive a motion for summary judgment by resting on the mere allegations of its pleadings. *See McCallum Highlands, Ltd. v. Washington Capital Dus, Inc.*, 66 F.3d 89, 92 (5th Cir. 1995). The nonmovant must go beyond the pleadings and

designate specific facts showing that there is a genuine issue for trial. *See Little*, 37 F.3d at 1075 (citing *Celotex*, 477 U.S. at 325). The nonmovant must “do more than simply show that there is some metaphysical doubt as to the material facts.” *Webb v. Cardiothoracic Surgery Assocs.*, 139 F.3d 532, 536 (5th Cir. 1998) (citing *Matsushita Elec. Indus. Co. v. Zenith Radio Corp.*, 475 U.S. 574, 586-87 (1986)).

“[W]hen a district court denies a motion for summary judgment on the basis that there exist genuine issues of material fact, the district court is actually making two separate conclusions: ‘First, the court has concluded that the issues of fact in question are genuine, *i.e.*, the evidence is sufficient to permit a reasonable factfinder to return a verdict for the nonmoving party. Second, the court has concluded that the issues of fact are material, *i.e.*, resolution of the issues might affect the outcome of the suit under governing law.’” *Lemoine v. New Horizons Ranch & Ctr., Inc.*, 174 F.3d 629, 633 (5th Cir. 1999) (quoting *Colston v. Barnhart*, 146 F.3d 282, 284 (5th Cir. 1998)); *Conoco, Inc. v. Medic Systems, Inc.*, 259 F.3d 369, 371 (5th Cir. 2001).

In deciding a summary judgment motion, “[t]he evidence of the nonmovant is to be believed, and all justifiable inferences are to be drawn in his favor.” *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 255 (1986). Credibility determinations are not part of the summary judgment analysis. *Id.* at 247-49. “Rule

56 ‘mandates the entry of summary judgment, after adequate time for discovery, and upon motion, against a party who fails to make a showing sufficient to establish the existence of an element essential to that party’s case, and on which that party will bear the burden of proof at trial.’” *Little*, 37 F.3d at 1075 (quoting *Celotex*, 477 U.S. at 322).

III. Analysis

A. The Motion to Strike the Peterson Declaration

Intergraph seeks to strike the Peterson declaration under Federal Rule of Civil Procedure 37(c)(1), which provides:

A party that without substantial justification fails to disclose information required by Rule 26(a) or 26(e)(1), or to amend a prior response to discovery as required by Rule 26(e)(2), is not, unless such failure is harmless, permitted to use as evidence at a trial, at a hearing, or on a motion any witness or information not so disclosed.

Under Federal Rule of Civil Procedure 26(a)(2), the parties were required to provide expert reports when they designated their experts.⁵ The issue is whether American

⁵ Rule 26(a)(2)(B) provides: “Except as otherwise stipulated or directed by the court, this disclosure shall, with respect to a witness who is retained or specially employed to provide expert testimony in the case or whose duties as an employee of the party regularly involve giving expert testimony, be accompanied by a written report prepared and signed by the witness. The report shall contain a complete statement of all opinions to be expressed and the basis and reasons therefor; the data or other information considered by the witness in forming the opinions; any exhibits to be used as a summary of or support for the opinions; the qualifications of the witness, including a list of all publications authored by the witness within the preceding ten years; the compensation to be paid for the study and testimony; and a listing of any other cases in which the witness has testified as an

Imaging failed timely to disclose or supplement Peterson's expert report on infringement.

This court initially set the cutoff for expert designations as May 15, 2003 and set the cutoff for rebuttal expert designations as May 24, 2003. (Docket Entry No. 86). At the parties' request, this court extended the cutoff for expert designations to May 30, 2003 and the cutoff for rebuttal expert designations to June 27, 2003. (Docket Entry No. 101). This court set a discovery deadline of July 25, 2003 and a docket call date of September 26, 2003. (*Id.*).

Before the discovery cutoff, Peterson submitted written expert reports dated March 19, 2003, May 30, 2003, and June 27, 2003. (Docket Entry No. 133, Exs. 5-7). Only the May 30 report specifically addressed whether the I/RAS software infringed the '393 Patent. In that report, Peterson stated that he believed I/RAS infringed the '393 Patent for several reasons, including:

- MicroStation and I/RAS B are two logically separate and distinct programs, which load separately into computer memory;
- MicroStation deals virtually exclusively with vector data and leaves the raster/pixel data to the I/RAS B package;

expert at trial or by deposition within the preceding four years.”

- MicroStation and I/RAS B coordinate their vector and raster operations, to the extent that they do, by communicating through a mechanism involving the use of the interrupt vector table in addition to and association with other similarly structured vector tables; and
- at least some of the disputed Intergraph products relocate and overstore interrupt vector locations and redirect interrupts in coordinating the display of the independent raster and vector images.

(Docket Entry No. 133, Ex. 6, May 30, 2003 Report of Gordon Peterson, ¶ 21). None of Peterson's expert reports submitted before the discovery deadline contained any claim infringement charts.

American Imaging submitted another declaration by Peterson in its response to Intergraph's motion for summary judgment filed on August 20, 2003, almost a month after the discovery cutoff and just over a month before the date set for docket call. (Docket Entry No. 124, Ex. B). In this declaration, Peterson provided his most detailed infringement analysis, including a claim chart. (*Id.* at Ex. B-4). Intergraph contends that Peterson's declaration was untimely filed under Rules 26(a)(2)(B) and 37(c)(1) and should be stricken.

A court has broad discretion to "preserve the integrity and purpose of the pretrial order." *Barrett v. Atl. Richfield Co.*, 95 F.3d 375, 380 (5th Cir. 1996) (quoting *Geiserman v. MacDonald*, 893 F.2d 787, 790 (5th Cir. 1990)). Rule 26

requires a “complete statement of all opinions to be expressed.” Rule 37(c)(1) does not permit a party to introduce evidence that it failed timely to disclose under Rule 26(a), without substantial justification. Federal Rule of Civil Procedure 37(b)(2)(B) allows a district court to refuse to allow a party failing to introduce materials not produced as required by Rule 26 or Rule 37 into evidence. When a district court strikes an expert’s testimony under Rule 37(b)(2)(B), the court must consider the following four factors:

- (1) the explanation, if any, for the party’s failure to comply with the discovery order;
- (2) the prejudice to the opposing party of allowing the witness to testify;
- (3) the possibility of curing such prejudice by granting a continuance; and
- (4) the importance of the witness’s testimony.

Barrett, 95 F.3d at 380; *Sierra Club, Lone Star Chapter v. Cedar Point Oil Co.*, 73 F.3d 546, 572 (5th Cir. 1996).

American Imaging provided no explanation as to why Peterson failed to produce an earlier expert report addressing the issue of infringement in detail. Peterson stated in his May 30, 2003 report that because he was unable to obtain the I/RAS source code, he “reserved the right to supplement and provide additional bases

for [his] opinions and findings of infringement. . . .” (Docket Entry No. 133, Ex. 6 at ¶ 21). At the hearing on August 26, 2003, Intergraph stated that it made available to American Imaging the source code for the I/RAS products. The only source code Intergraph did not make available to American Imaging was for the MicroStation software that Bentley controlled. Intergraph’s own experts did not have access to this source code. Intergraph explained, and American Imaging did not dispute, that other evidence in the summary judgment record, available to both parties, made the source code for the MicroStation software unnecessary to the infringement analysis.

This case has been litigated since 1997. American Imaging has prosecuted this infringement litigation for six years, but only provided its expert’s detailed report on infringement after the deadline for designating experts and the discovery deadline, and only a month before the docket call for the case. American Imaging has failed to explain its delay in providing Peterson’s detailed expert report on infringement.

American Imaging’s delay in producing Peterson’s expert report on infringement would materially prejudice Intergraph. American Imaging filed this suit for infringement six years ago. American Imaging produced Peterson’s expert report after discovery closed and after Intergraph moved for summary judgment, a month before docket call. *See Sierra Club*, 73 F.3d at 573 (defendant’s one-month delay in

producing expert witness disclosures caused plaintiff prejudice despite having been produced two months before trial). Although a continuance might cure any prejudice to Intergraph, this case has been in litigation for six years; a continuance would only result in further delay. This court has already extended the deadline for designation of experts in this case. (Docket Entry No. 101). “A continuance does not, in and of itself, ‘deter future dilatory behavior, nor serve to enforce local rules or court imposed scheduling orders.’” *Barrett*, 95 F.3d at 381 (quoting *Geiserman*, 893 F.2d at 792).

Without Peterson’s latest report, American Imaging clearly cannot raise a disputed fact issue as to infringement. Nevertheless, “even granting that the expert testimony was significant ‘the importance of such proposed testimony cannot singularly override the enforcement of local rules and scheduling orders.’” *Id.* (quoting *Geiserman*, 893 F.2d at 792). In *Barrett*, despite the fact that the four opinions the district court struck as untimely were important to that case, the Fifth Circuit upheld the district court’s decision, stating that “[a]ppellants’ repeated dilatory behavior even in the face of explicit warnings and the apparent inability of the experts to produce relevant opinions within the specified time frame renders hollow any claims of unfair prejudice.” 95 F.3d at 182.

In responding to the parties' joint motion to extend dates for expert designation and completion of discovery, this court stated that "[t]his motion is granted only to a limited extent. The amount of extra time the parties requested is excessive in light of the age and history of this case." (Docket Entry No. 101). The parties were on notice that this court expected compliance with the discovery orders due to the age of this case. Nevertheless, American Imaging, the plaintiff in this infringement action, failed timely to produce Peterson's expert report on infringement and its associated claim charts, waiting until after the expert report disclosure date, and after the discovery and motion cutoff dates.

In *Brumley v. Pfizer, Inc.*, 200 F.R.D. 596, 603-604 (S.D. Tex. 2001), plaintiffs submitted a supplemental affidavit from an expert whose report had previously been produced under Rule 26(a)(2)(B). The supplemental affidavit added new opinions to those the expert previously expressed in his Rule 26 report. The court excluded the supplemental affidavit containing the new opinions on the ground that Rule 26 requires "a complete statement of all opinions to be expressed." Similarly, Peterson's first three expert reports did not contain any detailed infringement analysis; he produced this only in response to Intergraph's motion for summary judgment.

This court GRANTS Intergraph's motion to exclude Peterson's declaration dated August 19, 2003 and produced in American Imaging's response to Intergraph's motion for summary judgment, filed August 20, 2003. In the alternative, this court considers the substance of the Peterson declaration of August 19, 2003 in analyzing whether a material fact issue exists as to the question of infringement.

B. Literal Infringement

The parties have stipulated that the elements of claims 12, 35, 36, and 42 are in means-plus-function format under 35 U.S.C. § 112, paragraph 6. The parties have also stipulated to the construction of these claims. (Docket Entry No. 102).

To find literal infringement of claim limitations written in means-plus-function format, a court must find, at a minimum, identity of the relationship between the claimed function and that of the accused device. *Telemac Cellular Corp. v. Topp Telecom, Inc.*, 247 F.3d 1316, 1332 (Fed. Cir. 2001) (citing *WMS Gaming, Inc. v. Int'l Game Tech.*, 184 F.3d 1339, 1350 (Fed. Cir. 2001)). Next, the court must satisfy itself that the accused device incorporates the same or equivalent structure to what the specification describes as performing that function. *Id.*

American Imaging contends that the I/RAS function `mdlView_setFunction(UPDATE_PRE, irasb_hook_update)` performs the identical

function as the “driver identification means,” “relocation means,” and “remapping means” of claims 12, 35, and 42. Claims 12 and 35 claim a “driver identification means for identification of existence of a display driver interrupt address in the vector table,” a “relocation means for relocating a CAD generated display driver interrupt from said display driver interrupt address to a clear location,” and a “remapping means for remapping a link program means to said display driver interrupt address.”

American Imaging argues that mdlView_setFunction inserts the I/RAS raster editor between the MicroStation CAD system and the display driver in the same way that the claimed invention inserts the link program between the CAD system and the display driver. The MicroStation CAD system contains a table of functions (the “user exit function table”) that the CAD system can call when different events or user inputs occur and a table of memory addresses for these functions. (Peterson Decl., ¶ 24; Entine Decl., ¶ 16). According to Gordon Peterson, American Imaging’s computer software expert, the entries in the table are indexed by numeric variables corresponding to different events, such as a screen update (UPDATE_PRE) and cursor motion on the screen (VIEW_MOTION). (*Id.* at ¶ 24, Ex. 14 at B06696). In the default condition, the entry indexed by UPDATE_PRE, a numeric variable, contains the address for the display driver interrupt routine or a return instruction to another routine which includes the display driver. (*Id.* at ¶ 27). The function

mdlView_setFunction(UPDATE_PRE, irasb_hook_update), a MicroStation CAD system routine called by I/RAS, replaces the default address of the display driver interrupt routine in the user exit function table with the address of the function irasb_hook_update, the raster editor. (*Id.* at ¶ 31). This makes the MicroStation CAD system call the I/RAS raster editor rather than the default display driver when a screen update event occurs. Peterson states that this is equivalent to the “remapping means” of the ‘393 Patent, which replaces the display driver interrupt address in the interrupt vector table of the operation system with the link program address.

Peterson states that mdlView_setFunction(UPDATE_PRE, irasb_hook_update) uses the value of the MicroStation CAD system variable UPDATE_PRE to identify the location of the display driver interrupt routine—or a return instruction that returns the machine to other routines that lead to the display driver interrupt routine—which is called when a screen update occurs. (Docket Entry No. 124, Ex. B, Declaration of Gordon Peterson, (“Peterson Decl.”), ¶ 24). The function mdlView_setFunction returns as output a pointer to the function that the MicroStation CAD system previously called in an update screen event. In the default case, the function called in an update screen event is the display driver interrupt

routine or a routine that contains a call to the display driver routine.⁶ (Peterson Decl., Ex. 13 at B06696). Peterson contends that this step is equivalent to the “driver identification” of the ‘393 Patent because the MicroStation CAD system is obtaining the address of the display driver interrupt routine.

Peterson states that I/RAS “relocates” the display driver interrupt routine when I/RAS replaces the display driver as the software to call in a screen update event. According to Peterson, I/RAS accomplishes the “relocation” of the display driver by passing to the function `irasb_hook_update` the pointer to the function that previously occupied the address. The function `irasb_hook_update` contains calls to MicroStation functions, such as `mdlWindow_lineDraw`, which control the display driver. (*Id.* at ¶ 28). Peterson asserts that this step is equivalent to the “relocation means” of the ‘393 Patent because the display driver interrupt routine is effectively “relocated” to the `irasb_hook_update`. (*Id.*).

I/RAS employs “dynamic linking” directly to communicate with MicroStation subroutines that send images to the computer screen. I/RAS subroutines can call MicroStation subroutines, and vice versa. When the computer screen image is to be updated, MicroStation calls I/RAS directly to run the subroutine

⁶ A pointer is a variable that stores the address of a variable containing data or a function in memory. It does not store data itself, but rather “points” to a location in memory containing data.

that determines which portion of the stored raster image is to be displayed. To send raster images for display, I/RAS calls functions within MicroStation, which drive the display device.

There are several critical differences between the claimed invention and the accused device. Unlike the claimed invention, the I/RAS software does not operate by identifying the display driver interrupt address in the interrupt vector table of the computer's operating system, moving that address to a new location in the vector table, and inserting a link program in the address previously occupied by the display driver. The I/RAS software does not manipulate a computer's operating system's interrupt vector table to communicate with the MicroStation CAD system, as would occur under claims 12 and 35.

I/RAS does not have to identify a driver display interrupt to replace. Display driver interrupts have values in the interrupt vector table which can be found, for example, by reading data from the display driver's configuration file. (Docket Entry No. 113, Declaration of David Chassin ("Chassin Decl."), ¶ 17). The user can relocate display driver interrupts from their default positions.⁷ (*Id.* at ¶ 18). To insert itself into the program flow, I/RAS calls the MicroStation CAD system's

⁷ An operating system reserves several interrupt vector table spaces for device generated interrupts. (Docket Entry No. 72, Ex. F, Abraham Silberschatz, OPERATING SYSTEM CONCEPTS, 5th Ed. 404 (Addison Wesley Longman 1998)).

mdlView_setFunction(UPDATE_PRE, irasb_hook_update) function, which places the function irasb_hook_update in the location of the user exit function table indexed by the variable UPDATE_PRE. (Entine Decl., Ex. 3 at B06696). It is true that the function mdlView_setFunction(UPDATE_PRE, irasb_hook_update) returns a pointer that identifies the address of the function previously called when an UPDATE_PRE event occurred. The function mdlView_setFunction(UPDATE_PRE, irasb_hook_update) does not, however, have to identify the previously-called function as a driver display.⁸ Nor does mdlView_setFunction(UPDATE_PRE, irasb_hook_update) have to make any identification for the purpose of replacing the function. The function irasb_hook_update replaces whatever function was present in the exit routine table position indexed by UPDATE_PRE. Although Peterson states that the default function in UPDATE_PRE is a display driver interrupt routine, or a routine that calls to the display driver interrupt routine, the function in UPDATE_PRE need not be the display driver.⁹ When a screen update event occurs,

⁸ Although irasb_hook_update(UPDATE_PRE, irasb_hook_update) returns a pointer to the previous function that the MicroStation CAD system called when an UPDATE_PRE event occurred, thereby “identifying” that function, it does not appear that the function irasb_hook_update relies on this identification to communicate with the display driver. The function irasb_hook_update does not take the pointer to that previous function and use it in calling the mdlWindow_lineDraw function. (Entine Decl., Ex. 3 at ING-025388.103; Peterson Decl., Ex. 13 at B06315).

⁹ Peterson noted that the returned pointer from the mdlView_setFunction(UPDATE_PRE, irasb_hook_update) function could point to any other interrupt routine to chain the interrupt routines. (Peterson Decl. at ¶ 25).

the MicroStation CAD system looks at the memory address containing the `irasb_hook_update` function. I/RAS does not literally infringe on the “driver identification means” element of claims 12, 35, and 42.¹⁰

I/RAS does not have to “relocate” the display driver interrupt routine when `irasb_hook_update` replaces the display driver interrupt routine as the routine the MicroStation CAD system calls in a screen update event. I/RAS simply makes the MicroStation CAD system look at a different address containing `irasb_hook_update`, the raster editor. The driver display interrupt routine is not “relocated” to an empty user interrupt, as in the ‘393 Patent. The I/RAS raster editor function `irasb_hook_update` simply replaces whatever function was present in the `UPDATE_PRE` position of the user exit routine table, whether it is the display driver function or some other function. The function `irasb_hook_update` contains direct calls to the MicroStation CAD system’s drawing routines, such as `mdlWindow_lineDraw`. It does not have to go through the user exit function table and rearrange addresses to call the MicroStation drawing functions, which in turn

¹⁰ Peterson contends that there are two places within the MicroStation CAD system’s code where the display driver identifies itself and other display driver interrupts in the operating system’s interrupt vector table and relocates and remaps these interrupts. (Peterson Decl., ¶ 33). He does not state, however, that this remapping and relocation is done in conjunction with the insertion of I/RAS in the program flow. These code sections are in MicroStation’s source code, not I/RAS’s source code. I/RAS uses these code sections only when it refers to “some” MicroStation routines, which Peterson did not specify. (Peterson Decl., ¶ 33).

ultimately call the display driver interrupt routine. No relocation means is needed to enable communication with the display driver interrupt routine. I/RAS does not literally infringe the “relocation means” element of claims 12, 35, and 42.

I/RAS does map the raster editor routine to the location in the user exit function table previously occupied by another routine, which could be the display driver routine. I/RAS’s mapping, however, does not occur in the operating system’s interrupt vector table. Claims 12 and 35 involve an “operating system having an interrupt vector table” upon which the “driver identification means,” the “relocation means,” and the “remapping means” all act. The parties have stipulated to the following construction of the term “operating system having an interrupt vector table”:

an operating system such as PC/DOS or MS/DOS in the context of the ‘393 Patent which uses interrupt vector tables.

(Docket Entry No. 102, ¶ 1). No “remapping” occurs in the operating system’s vector table. I/RAS does not literally infringe the “remapping means” element of claims 12 and 35.

All of I/RAS’s output goes to software routines in MicroStation. I/RAS does not directly control the display device and does not issue commands directly to the display device. (Entine Decl., ¶ 21). I/RAS does not determine the address of the

display driver interrupt, does not relocate the display driver interrupt in the interrupt vector table, and does not map itself to the display driver interrupt's original address. These steps are unnecessary, because I/RAS can communicate directly with MicroStation. Because I/RAS does not perform the functions of identifying a display driver interrupt, relocating the driver display interrupt, and remapping itself to the display driver interrupt's original interrupt address, I/RAS does not literally infringe claims 12 and 35 of the '393 Patent. *See Interactive Pictures Corp. v. Infinite Pictures, Inc.*, 274 F.3d 1371, 1382 (Fed. Cir. 2001)(finding no literal infringement where allegedly infringing signal processing element used different input to create same output as claimed element; because inputs were different, the function of the claimed element and the allegedly infringing element were not identical).

I/RAS does not literally infringe claims 36 and 42 because it lacks the identical function of the claimed means. One of the elements of claim 36 is an "intercepting means for intercepting a display driver interrupt generated by said CAD system in response to a user input command." I/RAS does not intercept display driver interrupts. The function `irasb_hook_update` is called directly by the MicroStation CAD system when an `UPDATE_PRE` event occurs, and directly calls MicroSystem's display driver routines. The elements of claim 42 include a "driver identification means for identification of existence of a display driver driven by the

CAD system,” “relocation means for relocating a display driver identified by said driver identification means from a driver address to a clear location therefor,” and “remapping means for remapping the edit program means to said driver address.” Claim 42 is not limited to the use of an operating system containing an interrupt vector table. Nevertheless, I/RAS does not literally infringe claim 42, because it does not identify or relocate a display driver.

C. The Doctrine of Equivalents

A claim that does not literally read on an accused device may nevertheless be infringed under the doctrine of equivalents if the differences between the claim and the accused device are insubstantial. *WMS Gaming*, 184 F.3d at 1352-53 (citations omitted). Infringement under the doctrine of equivalents often turns on whether the accused and the patented component performing the substantially the same function, in substantially the same way, with substantially the same result. *Telemac Cellular*, 247 F.3d at 1332 (citing *Unidynamics Corp. v. Automatic Prod. Corp. Int’l, Ltd.*, 157 F.3d 1311, 1322 (Fed. Cir. 1998)). The application of the doctrine of equivalents is an objective inquiry made on a limitation-by-limitation basis. *Unidynamics*, 157 F.3d at 1322 (citing *Warner-Jenkinson Co. v. Hilton Davis Chem. Co.*, 520 U.S. 17, 40, 117 S. Ct. 1040, 1054, 137 L. Ed. 2d 146 (1997)).

American Imaging contends that fact issues remain as to whether I/RAS infringes the remaining claims of the '393 Patent. American Imaging asserts that the MicroStation CAD system's user exit routine table, which I/RAS manipulates, is substantially the same as the interrupt vector table that the '393 Patent manipulates.

The I/RAS software does not function in substantially the same way as the invention claimed in the '393 Patent, viewed on a limitation-by-limitation basis. I/RAS does not make any changes to the interrupt vector table of the host computer's operating system.¹¹ The I/RAS software does not identify a display driver for the purpose of replacing it. It does not require a mechanism for identifying programs when it replaces the existing function in the UPDATE_PRE section of the user exit function table with the function irasb_hook_update. Although the function that places irasb_hook_update in the UPDATE_PRE section returns a pointer to the previous function, it does not have to be identified as a display driver. In other words, the claimed invention's "driver identification means" and the accused device's I/RAS insertion program operate differently. The claimed invention identifies the position occupied by the display driver interrupt routine and then relocates it and puts the raster editor in its place. The accused device removes whatever is present in the

¹¹ According to Intergraph, Microsoft, a major manufacturer of operating systems, no longer allows users to manipulate the interrupt vector tables of its operating systems. American Imaging does not dispute this assertion.

UPDATE_PRE position of the user exit routine table, without identifying it as a display driver for the replacement process, and assigns a pointer to that routine.

The accused device does not require relocating the display driver address to a new position in the user exit routine table to send output to the screen. The accused device returns a pointer to the address of the function previously occupying the UPDATE_PRE position in the user exit routine address, but the function irasb_hook_update does not rely on this address to send output to the display driver routine. The irasb_hook_update calls the drawing routines directly, which the “dynamic linking” approach of the accused device allows.

The overall approach taken by I/RAS differs substantially from the approach taken by the ‘393 Patent. By integrating the I/RAS software and MicroStation CAD system, the accused device eliminated the need for identifying display drivers and reordering the interrupt vector table of the host computer. The accused device allows the I/RAS software directly to access and call routines in the MicroStation CAD system, including the display driver.

The fact that the ‘393 Patent manipulates an operating system’s interrupt vector table was important to the Federal Circuit’s finding that the record did not support invalidation of the remaining claims of the ‘393 Patent on obviousness grounds. The Federal Circuit stated that “the record lacks any other evidence

regarding whether SuperPaint . . . suggests to one of ordinary skill in the art to reconfigure Super Paint to enable it to work with an operating system having an interrupt vector table.” (Docket Entry No. 58, p. 13). I/RAS does not manipulate an operating system’s interrupt vector table, but rather applies dynamic linking between the I/RAS and MicroStation routines.

The undisputed differences between the I/RAS software and the ‘393 Patent lead to the conclusion that, as a matter of law, the doctrine of equivalents does not apply.

IV. Conclusion

Intergraph’s motion to strike Gordon Peterson’s August 19, 2003 declaration is GRANTED. Alternatively, considering Peterson’s declaration as part of the summary judgment record, American Imaging has failed to show that a question of material fact exists as to whether Intergraph’s I/RAS software infringes the ‘393 Patent. Intergraph’s motion for summary judgment of noninfringement is GRANTED.¹²

SIGNED on September 2, 2003, at Houston, Texas.



Lee H. Rosenthal
United States District Judge

¹² Because of the finding of noninfringement, this court does not reach the issue of whether claims 12, 35, 36, and 42 are invalid under sections 102(b), 102(g), and 103. Intergraph’s motions to strike the expert report of Dale Boisso (Docket Entry No. 116); to stay service of its rebuttal expert (Docket Entry No. 116); to enter a revised pretrial scheduling order (Docket Entry No. 116); and to amend its answer (Docket Entry No. 120) are MOOT.